OPTIMIZATION OF SELECTED REMOTE SENSING ALGORITHMS FOR EMBEDDED NVIDIA KEPLER GPU ARCHITECTURE

Lubomír Říha, Jacqueline Le Moigne**, Tarek El-Ghazawi****

* IT4Innovations National Supercomputing Center, VŠB-Technical University of Ostrava, Ostrava, Czech Republic
** NASA Goddard Space Flight Center, Software Engineering Division, Greenbelt, MD, USA
*** High-Performance Computing Laboratory, The George Washington University, Ashburn, VA, USA

ABSTRACT

This paper evaluates the potential of embedded Graphic Processing Units in the Nvidia's Tegra K1 for onboard processing. The performance is compared to a general purpose multi-core CPU and full fledge GPU accelerator. This study uses two algorithms: Wavelet Spectral Dimension Reduction of Hyperspectral Imagery and Automated Cloud-Cover Assessment (ACCA) Algorithm. Tegra K1 achieved 51% for ACCA algorithm and 20% for the dimension reduction algorithm, as compared to the performance of the high-end 8-core server Intel Xeon CPU with 13.5 times higher power consumption.

Index Terms— remote sensing, GPU, Tegra K1, ACCA, dimension reduction

1. INTRODUCTION

This paper evaluates the suitability of new embedded Graphic Processing Units (GPU) in the Nvidia's Tegra K1 (K1) System-on-Chip (SoC) with typical Typical Design Power (TDP) under 7W [1] for onboard processing. The performance of this SoC is compared to two modern High Performance Computing (HPC) architectures: (1) a general purpose multi-core CPU (8-core Sandy Bridge E5-2470, 2.3GHz, TDP 95W [2]) and (2) GPU accelerator (Nvidia Tesla K20 (K20), TDP 225W [3]). For this study, we selected two algorithms:

1. Wavelet Spectral Dimension Reduction of Hyperspectral Imagery: The principle of this method is to apply a discrete wavelet transform to hyperspectral data in the spectral domain and at each pixel location. The optimal level of wavelet decomposition is computed adaptively for each pixel. See [4] for more details.
2. Automated Cloud-Cover Assessment (ACCA) Algorithm: The ACCA algorithm determines and rates the overall cloud cover of an image through 2 steps: Pass-One isolates clouds from non clouds by utilizing eight threshold-based filters, then Pass-Two resolves the detection ambiguities from Pass-One by computing global statistics over the image. See [5] for more details.

This paper shows that the performance achieved using this new SoC designed for battery powered devices is comparable to HPC hardware with significantly higher power consumption.

2. HARDWARE ARCHITECTURES

The Intel Xeon Sandy Bridge CPU is a general purpose processor designed to handle a wide variety of workloads. It has a small number, up to 8, of high performance cores with 64 bit wide SIMD (Single Instruction Multiple Data) units and large on-chip caches (~22 MB) designed to minimize the effect of limited memory bandwidth (38.4 GB/s).

	Nvidia Tegra K1 (GPU part)	**8-core Intel Sandy Bridge E5-2470**	**Nvidia Tesla K20 GPU**
Architecture type	embedded SoC with Kepler GPU	general purpose CPU for HPC	GPU accelerator for HPC
Frequency	0.852 GHz – GPU part	2.3GHz	0.706GHz
Number of Cores	192 SP scalar cores – GPU part	64 SP / 32 DP cores (8 SIMD cores)	2496 SP / 832 DP scalar cores
On-Chip Caches	64 KB L1 per 192 SP cores 128KB L2 per chip	32+32KB L1, 256 KB L2 per SIMD core 20 MB L3 per chip	64 KB L1 per 192 SP cores; 1536KB L2 per chip
SIMD width	32 for both SP and DP	8 for SP and 4 for DP	32 for both SP and DP
Peak Performance	327 SP / 13 DP GFLOPS	147 SP / 74 DP GFLOPS	3524SP/1160DP GFLOPS
Mem. Size; Bandwidth	2GB at Jetson TK1; 14.9 GBPS	up to 384 GB; 38.4 GBPS	5GB; 208 GBPS
TDP	7W (SoC + DRSM only)	95W (CPU only)	225W (accelerator only)

Table 1. Main parameters of the selected hardware architectures

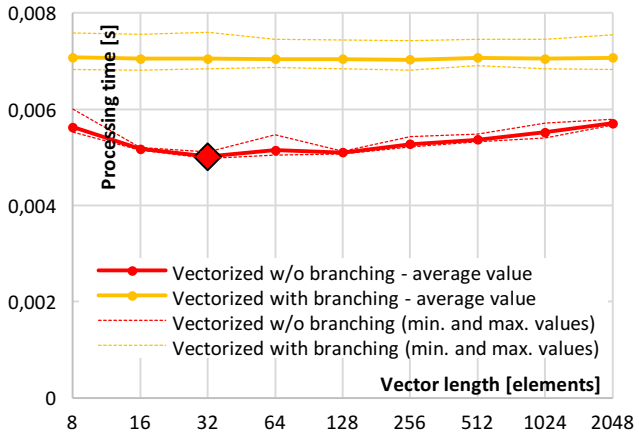


Fig. 1. Optimal vector length for the ACCA algorithm running on CPU is 32.

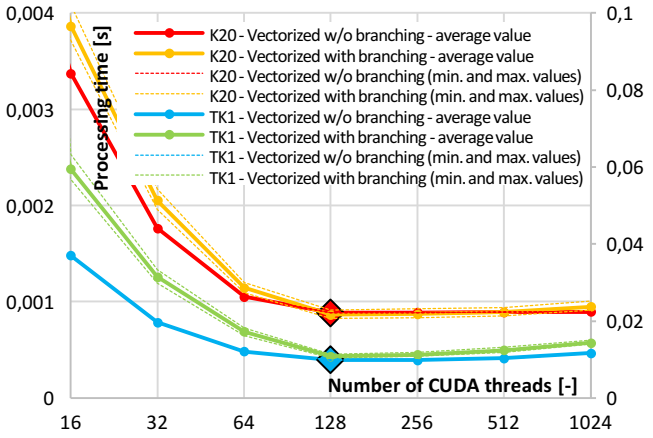


Fig. 2. Optimal number of threads per block for ACCA on Tesla K20 and Tegra K1 is 128.

The performance is achieved by the SIMD units which can process 4 Double Precision (DP) or 8 Single Precision (SP) values per core per clock cycle, or 32 DP/64 SP operations per clock per chip. For more details see Table 1.

Both K20 and the GPU part of K1 are based on the NVidia Kepler architecture. Designed as throughput architecture, it has small L1 and L2 caches but fast memory interface. Processing cores are organized in Streaming Multiprocessors (SMX). Each SMX has 6 groups of 32 SP cores (192 cores total), that can be seen as 6 SIMD units. K20 has additional 64 DP cores, but not K1. The K20 contains 13 SMXs while K1 only 1 SMX. In terms of peak performance in SP the K1 is 10.7 times slower than K20, but 2.2 times faster than the CPU. In terms of memory bandwidth K1 is 13.9 times slower than K20 and 2.5 times slower than the CPU. But K1 is a SoC designed for mobile and embedded systems with low power consumption. The entire Jetson TK1 development board [6] consumes ~12.5W (SoC + DRAM ~7W) under full load.

3. IMPLEMENTATION AND OPTIMIZATION

The main focus of the optimization part is to explore techniques that allow efficient utilization of the parallel hardware. Even though the architectures are different, they all use SIMD units. This means that data is processed as short vectors where identical operations are executed on all elements. The number of elements per vector, or SIMD width, is 4 DP or 8 SP values for CPU and 32 SP/DP for GPUs, see Table 1. The efficiency of vector processing is significantly reduced by the branching in the code caused by conditional statements.

Both algorithms have a very high degree of parallelism defined by the number of pixels and can be efficiently vectorized. In case of dimension reduction algorithm, the vectorization is used across spectral bands within a pixel, when computing wavelet coefficients, while pixels are processed independently. There are no conditional statements, no branching, within the vectorized section of the code that can reduce vectorization efficiency. For each pixel, the algorithm requires fast access to original data, reduced data and reconstructed data as these are accessed multiple times. Therefore it is very efficient to keep it inside the on-chip caches. This can be achieved in case of the CPU, but K20 and K1 do not have enough on-chip storage which results in performance penalty. This algorithm was chosen to evaluate the effect of K1's small caches on the performance.

On the other hand, the ACCA algorithm contains a large number of conditional statements (one for each threshold-based filter), controlled by an input data. This results in a significant processing time variation depending on cloud coverage, see Figure 3 left. This is a problem for on-board real-time processing systems. To minimize this effect and also to maximize the performance of the SIMD units, two new versions of the ACCA algorithm were developed: (1) Vectorized without Branching (VNB) and (2) Vectorized with Branching (VWB). The VWB algorithm can utilize SIMD units but still exhibits processing time variation. In the VNB algorithm all threshold-based filters are redesigned to avoid branching by setting specific bits of a register. At the end, the register contains the value describing whether the pixel is a cloud or not. This means that all filters are executed for every pixel (this is not the case of the original ACCA algorithm), which creates more work. But since this workload can be processed much more efficient by the SIMD units, the VNB algorithm delivers faster processing. In the case of Tegra K1, the processing time variation as a function of cloud coverage is reduced from ~9% to 0.2%, and the processing time itself is reduced by 8.3% when compared to the VWB version.

The performance of the proposed algorithms also depends on the length of the vector processed per SIMD unit. In case of the CPU this parameter is called vector length. In CUDA model for GPUs the same parameter is described by number of threads per block. See Figure 1 and 2 for optimal configuration for all three architectures. These figures also show the variation for different vector lengths.

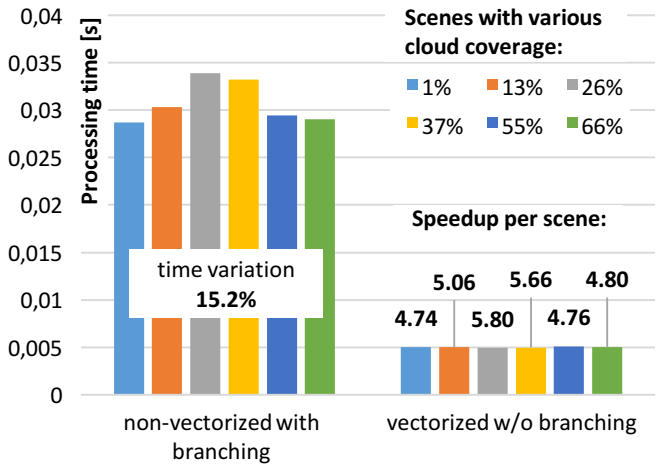


Fig. 3. Speedup achieved by vectorization for CPU is between 4.7 and 5.8. The processing time variation of original algorithm for different input data is 15.2%. The values above the bars show the speedup for different scenes with various cloud coverage from 1% to 66%.

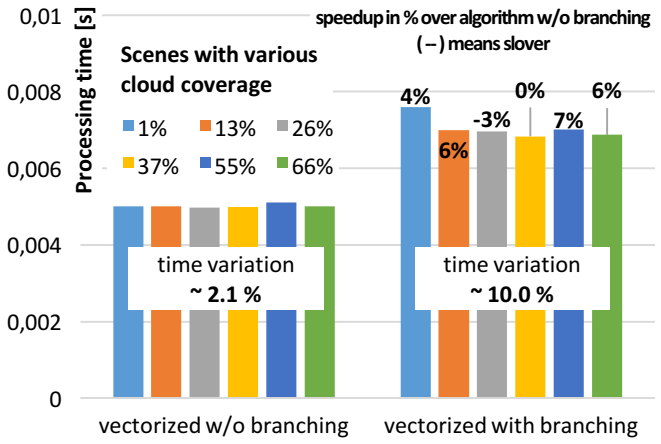
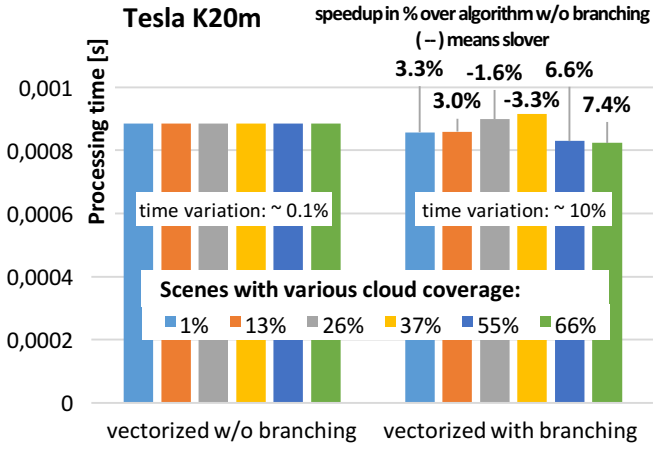


Fig. 4. Processing time variation based on input data with various cloud coverage 1%, 13%, 26%, 37%, 55% and 66% for CPU. The values above the bars describe the difference in processing time: negative values means slower than vectorized w/o branching algorithm.



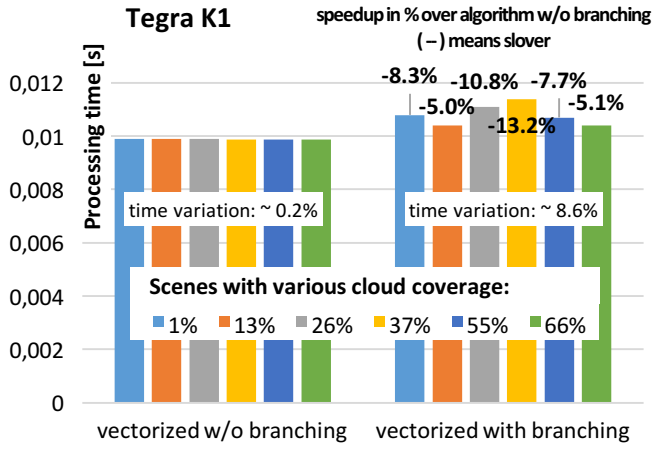


Fig. 5. Processing time variation based on input data with various cloud coverage 1%, 13%, 26%, 37%, 55% and 66% for Tesla K20 and Tegra K1 GPUs. The values above the bars describe the difference in processing time: negative values mean slower than vectorized no-branching algorithm. Image size is 2048x2048 pixels.

4. RESULTS

The proposed VNB version of the ACCA algorithm brings three major improvements: (1) enables the execution of the algorithm on the K20 and K1 GPUs; (2) significantly improves the performance, see speedup up to 5.7 for CPU in Figure 3, and (3) reduces the processing time variation for different scenes, from 15.2% of the original algorithm to 1.0% for CPU (see Figure 4), 0.1% for K20 and 0.2% for K1 (see Figure 5). In all figures, the 6 different colors represent datasets with cloud coverage varying between 1% and 66%.

The performance of the VNB version of the ACCA algorithm for all three architectures and all datasets is shown in Figure 6. The CPU is used as a baseline, with speedup equal to 1, and K1 and K20 are compared to it. As expected, the high performance K20 is on average 5.8 times faster, but the K1, with 13 times lower power consumption achieved 51% performance of the high-end 8-core server CPU.

The wavelet spectral dimension reduction algorithm performance on all three architectures is shown in Table 2. It was tested in a data independent fashion, so that all pixels are reduced by 4 wavelet decomposition levels. For each level the reconstruction to the original size and evaluation using a cross-correlation function is performed. Unlike ACCA this algorithm efficiently utilizes large CPU caches while the throughput GPU architecture is less efficient. This translates into the performance of K1 being about 20% of the CPU. Taking into account the 13 times lower power budget, the

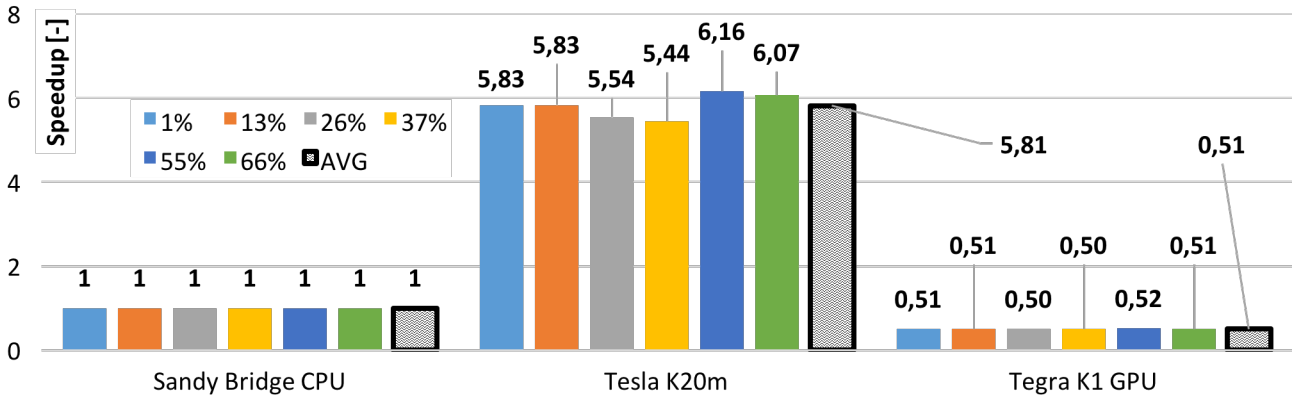


Fig. 6. Chip-to-chip performance comparison of the vectorized ACCA algorithm without branching for image size 2048x2048 pixels.

Architecture	Spectral Bands per pixel [-]	Processing time [s]	Performance [Mpix per second]	Speedup over CPU [-]
8-core CPU	128	0.0303	**3.14**	1
	256	0.0394	**2.42**	1
	512	0.0627	**1.52**	1
Nvidia Tesla K20	128	0.0118	**8.08**	**2.57**
	256	0.0177	**5.39**	**2.23**
	512	0.0411	**2.32**	**1.53**
Nvidia Tegra K1	128	0.6133	**0.61**	**0.20**
	256	0.4225	**0.42**	**0.19**
	512	0.1793	**0.18**	**0.17**

Table 2. Chip-to-chip performance comparison of the Wavelet Spectral Dimension Reduction algorithm for image size 100,000 pixels.

new Tegra K1 SoC has a great potential for onboard processing of complex algorithms.

5. CONCLUSIONS

This paper evaluates the feasibility of a new mobile many-core architecture, the 192-core GPU of the Tegra K1 SoC, for onboard processing, using two remote sensing algorithms. In order to gain optimal performance we had to redesign the original algorithms to support SIMD processing. Tegra K1 achieved (1) 51% for ACCA algorithm and (2) 20% for the dimension reduction algorithm, as compared to the performance of the high-end 8-core server Intel Xeon CPU. Both algorithms use only a GPU part of the SoC, leaving the 4+1 ARM Cortex A15 general-purpose cores available for other tasks.

6. REFERENCES

[1] Nvidia, "NVIDIA Tegra K1: A New Era in Mobile Computing", http://www.nvidia.com/content/PDF/tegra_white_papers/Tegra-K1-whitepaper-v1.0.pdf, 2014.

[2] Intel, "Intel® Xeon® Processor E5-2400 Product Family", http://www.intel.com/content/dam/www/public/us/en/documents/datasheets/xeon-e5-2400-vol-1-datasheet.pdf, May 2012.

[3] Nvidia, "NVIDIA's Next Generation CUDATM Compute Architecture: Kepler TM GK110", http://www.nvidia.com/content/PDF/kepler/NVIDIA-kepler-GK110-Architecture-Whitepaper.pdf, 2012.

[4] S. Kaewpijit, J. Le Moigne, T. El-Ghazawi, "Automatic reduction of hyperspectral imagery using wavelet spectral analysis," Geoscience and Remote Sensing, IEEE Transactions on , vol.41, no.4, pp.863,871, April 2003.

[5] R. R. Irish, et al. "Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm." Photogrammetric Engineering & Remote Sensing 72.10 (2006): 1179-1188.

[6] Nvidia, "NVIDIA Jetson TK1 Development Kit", http://developer.download.nvidia.com/embedded/jetson/TK1/docs/Jetson_platform_brief_May2014.pdf , 2014.